**Decisions and Coding Practices**

1. **Rest Assured for API Testing:**

* Built-in Support for HTTP Methods and Request Specifications.
* Handling Complex Responses:
* Support for BDD.
* Fluent API for Assertions

1. **Framework and Structure**:

The test suite is implemented using Cucumber with TestNG integration, which combines behavior-driven development (BDD). This setup allows non-technical stakeholders to understand the test scenarios.

1. **Base URL Configuration**:

The base URL is initialized from a properties file using BaseUrl.propInitiate();.

To make the tests adaptable to different environments without changing the code.

1. **Dynamic Endpoint Handling**:

he step definition for sending GET requests dynamically constructs the endpoint using parameters (@When("user send a GET request to : {string} {string}")). This allows for testing multiple endpoint combinations without hardcoding URLs, improving test coverage and reusability.

1. **Response Validation**:

* Status Code Check: Checking if the rest assured’s API is returning the expected status code.
* Non-Empty Response Check: Checking that the response isn’t null and not empty.
* Content Validation: checking that the response contains specific expected strings.

1. **Extent Report:**

* Detailed Step Logging.
* Enhanced Test Reporting and Visualization.
* Integration with TestNG and Cucumber:
* Ease of Sharing and Collaboration.